On the dynamic technician routing and scheduling problem

Victor Pillac, Christelle Gueret, Andrés Medaglia

On the dynamic technician routing and scheduling problem

Victor Pillac
École des Mines de Nantes, IRCCyN UMR 6597, Nantes, France
Universidad de los Andes, Industrial Engineering Department, Bogotá, Colombia
Email: vpillac@mines-nantes.fr

Christelle Guéret
École des Mines de Nantes, IRCCyN UMR 6597, Nantes, France
Email: gueret@mines-nantes.fr

Andrés L. Medaglia
Universidad de los Andes, Industrial Engineering Department, Bogotá, Colombia
Email: amedagli@uniandes.edu.co

1 Introduction

The Technician Routing and Scheduling Problem (TRSP) deals with a limited crew of technicians $\mathcal{K}$ that serves a set of requests $\mathcal{R}$. In the TRSP, each technician has a set of skills, tools, and spare parts, while requests require a subset of each. The problem is then to design a set of tours of minimal total duration such that each request is visited exactly once, within its time window, by a technician with the required skills, tools, and spare parts. The TRSP naturally arises in a wide range of applications, including telecoms, public utilities, and maintenance operations.

A distinctive feature of this problem is that it introduces several compatibility constraints between technicians and requests. While skills are intrinsic attributes, technicians may carry different tools and spare parts over the planning horizon. Technicians often start their tour from home, with a set of tools (renewable resources) and spare parts (consumed once the technician serves a request) that allows them to serve an initial set of requests. Technicians may have the opportunity to replenish their tools and spare parts at a central depot at any time to service more requests.

Despite its numerous practical applications and its challenging features, static technician routing and scheduling problems have received limited attention until recently. Xu and Chiu [1] studied a variant of the TRSP in which the objective is to maximize the number of requests served while accounting for skill constraints and request priorities. Tang et al. [2] considered priorities and formulated a multi-period maximum collection problem in which time-dependent rewards are granted for the completion of a request. Tsang and Voudouris [3] solved a problem faced by British Telecom where technician skills affect the time required to service a request.

On the other hand, a number of technological advances have led to a renewed interest in dynamic vehicle routing problems, leading to the development of new optimization approaches. We refer the interested reader to the recent survey by Pillac et al. [4] for a detailed review of such approaches.

To the best of our knowledge, no work considers neither tools, nor spare parts, nor the arrival of new requests, three important components of real-world applications. The present work addresses this aspect and proposes an optimization approach for the dynamic version of the problem, noted D-TRSP, where new requests arrive during the execution of the routes.

2 The proposed approach

The proposed approach is based on an parallel Adaptative Large Neighborhood Search (pALNS) algorithm which is used first to compute an initial solution, and then to reoptimize the solution whenever a new customer request arrives.

2.1 Computation of an initial solution

A regret heuristic [5] is first used to build an initial solution: at each iteration the algorithm inserts (at the best position) the request with the lowest *regret* value. The regret value is the difference between the cost of the first and second best insertions. It is a measure of how desirable it is to insert a request now that is used to prevent myopic decisions.

This initial solution is then improved by a pALNS algorithm. The ALNS algorithm [6] works by successively *destroying* (removing requests) and *repairing* (reinserting requests) a current solution. In the parallel version we propose, the algorithm maintains a pool of promising solutions that are optimized in p subprocesses. For each *master* iteration, a subset of p promising solutions is selected randomly and distributed to the available subprocesses. Then for l iterations, each subprocess selects destroy and repair operators with a roulette wheel that adaptively reflects their past performance. The resulting new solution is either accepted as the subprocess current solution or rejected. Finally, each subprocess reports the best solution found to the master process and the pool of promising solutions is updated. In addition, the weights of the destroy and repair operators are updated depending on their performance. The algorithm stops after n master iterations, which corresponds to $N = nl$ ALNS iterations. We adapted three destroy operators (*random*, *critical*, and *related*) and two repair heuristics (*best insertion* and *regret*).

2.2 Re-optimization procedure

Each time a new request arrives, we fix the currently executed portion of the solution, and re-run the pALNS for a limited number of iterations, leading to a reoptimized solution. If at the end of this procedure the new request could not be inserted, then we mark it as rejected.

3 Computational results

In this section we report computational results of the approach on 36 randomly generated instances of the D-TRSP, based on the Solomon [7] benchmark[1]. For each instance, we generated skill, tool, and spare part information, and associated a depot (home) to each of the 25 vehicles (technicians). Additionally, we generated release dates for a proportion of the requests, corresponding to the time at which they are made known to the algorithm.

We compare the proposed approach with a regret-2 heuristic, that consists in inserting new requests at the best possible position in the current solution, and rejecting requests that cannot be inserted, the initial solution being the same as the one used in the pALNS.

All experiments were run with a limit of $N = 25000$ iterations for the calculation of the initial solution, and $N = 5000$ for subsequent reoptimizations.

	pALNS		**regret-2**	
DOD	**VI** (%)	**R**	**VI** (%)	**R**
10	0.07	0.00	18.52	0.25
50	2.09	0.33	36.58	1.08
90	2.50	0.92	44.98	2.00
Average	1.55	0.42	33.36	1.11

Table 1: Computational results for 36 randomly generated D-TRSP instances.

Table 1 reports preliminary results for the 36 instances. The first column contains the degree of dynamism (DOD) defined as the proportion of dynamic requests. The second and third columns reports the average Value of Information [8] (VI)[2] and the average number of rejected requests (R) for the pALNS, while the fourth and fifth contains these statistics for the regret-2 heuristic. Running times are of 50s on average for the calculation of the initial solution and 11s for subsequent reoptimizations.

These results show that the proposed approach leads to better solutions both in terms of quality of the routing (measured by VI) and ability to cope with new requests (measured by R). In addition, we can note that the regret-2 approach rejects an average of 4.4 requests

[1] We used the following subset of 100-customers instances: {[R,C,RC][101,102,201,202]}

[2] The value of information for instance I is defined as the ratio $\frac{z(I)-z(I_{\text{off}})}{z(I_{\text{off}})}$ where $z(I)$ is the value of the solution found by the algorithm for the dynamic instance, and $z(I_{\text{off}})$ is the value of the solution for static instance I_{off} in which all requests are known beforehand, obtained running pALNS with 100k iterations.

in 9 of the 36 instances, while pALNS rejects an average of 3.0 requests on only 5 instances. This illustrates the robustness of pALNS that is less likely to take bad decisions that will then cause either the rejection of a higher number of requests, or result in a poor routing.

4 Conclusions and research perspectives

In this study we introduced a new challenging routing problem with numerous applications, namely a variant of the Technician Routing and Scheduling Problem. We proposed a metaheuristic based on a parallel ALNS algorithm fast enough to be used to solve the problem in a dynamic context. We validated and measured the performance of the proposed approach on randomly generated instances of the D-TRSP.

Future work will focus on considering additional strategies such as waiting or request buffering to improve the quality of the dynamic routing, and using stochastic knowledge gathered from historical data to better anticipate its execution.

Acknowledgements Financial support for this work was provided by the CPER (Contrat de Projet Etat Region) Vallée du Libre (France); and the Centro de Estudios Interdisciplinarios Básicos y Aplicados en Complejidad (CEIBA, Colombia). This support is gratefully acknowledged.

References

[1] J. Y. Xu and S. Y. Chiu. Effective heuristic procedures for a field technician scheduling problem. *Journal of Heuristics*, 7(5):495–509, 2001.

[2] E. Tang, H.and Miller-Hooks and R. Tomastik. Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research Part E: Logistics and Transportation Review*, 43(5):591–609, 2007.

[3] E. Tsang and C. Voudouris. Fast local search and guided local search and their application to british telecom's workforce scheduling problem. *Operations Research Letters*, 20(3):119 – 127, 1997.

[4] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. Technical report, CIRRELT, 2011. CIRRELT-2011-62.

[5] Jean-Yves Potvin and Jean-Marc Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331 – 340, 1993.

[6] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[7] M. M. Solomon. Algorithms for the vehicle-routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[8] K. Lund, O. B. G. Madsen, and J. M. Rygaard. Vehicle routing problems with varying degrees of dynamism. Technical report, IMM Institute of Mathematical Modelling, 1996.